Bilkent University
Department of Computer Engineering

Senior Design Project
T2439
Fennec Radio

Analysis and Requirement Report

22103782, Süleyman Yağız Başaran,
yagiz.basaran@ug.bilkent.edu.tr
22103559, Bartu Özyıldırım,
bartu.ozyildirim@ug.bilkent.edu.tr
22103766, Hikmet Kaan Oktay,
kaan.oktay@ug.bilkent.edu.tr
22101910, Feza Emir Çelik,
feza.celik@ug.bilkent.edu.tr
22003865, Atakan Keser,
atakan.keser@ug.bilkent.edu.tr

Supervisor: Özgür Ulusoy
Course Instructors: Atakan Erdem, Mert Bıçakçı

<09/12/2024>

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Contents

Analysis and Requirement Report

Fennec Radio

1 Introduction

Traditional radio is static, one-way, and lacks personalization. Listeners have limited control over what they hear, being confined to the schedules and content determined by radio stations. For content creators, the barriers to entry are significant, involving high costs and complex broadcasting technology. Aspiring broadcasters often need to apply to radio companies and meet strict-rich requirements, which can be discouraging.

In a world where audiences crave interactivity, customization, and real-time engagement, radio is struggling to keep up with modern user expectations. Therefore, **FennecRadio** enters the party.

There's a need for a platform that allows users to both broadcast and listen, while incorporating popular and advanced technology like AI to create an interactive, personalized radio experience. Whether it's sharing music, discussing news, or just casual chatting, everyone should have the freedom to create and consume content that resonates with them which is falling behind when the traditional radio is considered.

This project seeks to revolutionize the radio experience by making it more accessible, interactive, and personalized for everyone. Additionally, this report contains a brief description of the project, its architecture and technologies, project constraints, ethical issues, standards, functional and non-functional requirements, market research, academic analysis, and mockups.

2 Current System

The current radio landscape is dominated by traditional, static platforms that offer a one-way, non-interactive listening experience. Listeners have nearly no control over the content they consume, being restricted to the rigid schedules and limited variety set by radio stations. Customization options are virtually nonexistent, leaving audiences unable to tailor their experience to their personal preferences.

For content creators, the challenges are equally discouraging. Broadcasting on traditional platforms involves significant upfront costs, complex technology, advanced education and strict industry requirements. Aspiring creators often face a daunting process of applying to established radio companies, with limited opportunities for creative freedom or direct audience engagement. These high entry barriers limit innovation and exclude talented people from sharing their voices.

In an age where interactivity and personalization are the norm across digital platforms, traditional radio struggles to compete. Even though some competitors have introduced online streaming or podcast services, these solutions still fall short in providing the real-

time interaction and creative flexibility that the modern state of the entertainment sector demands.

This gap in the current system shows a clear need for innovation in the radio domain, which is an opportunity that **FennecRadio** aims to take advantage of. By using AI and advanced digital technologies, our platform addresses the limitations of traditional radio while empowering both listeners and broadcasters with a dynamic, customizable, and accessible experience.

3 Proposed System

3.1 Overview

The Fennec Radio system is an interactive online radio platform that combines user personalization, real-time broadcasts, and AI-driven content to deliver an engaging experience. **User authentication** ensures secure access through email or AWS Cognito, with email verification and password recovery simplifying account management. Users can customize their profiles by selecting usernames, updating them once every two months, and uploading profile or channel images for personalization.

On the **main page**, users can browse live broadcasts, search by genre, topic, or broadcaster, and follow channels for notifications. Channels can also be hidden for a tailored browsing experience. Within the **broadcast room**, listeners can join streams, adjust audio settings, interact through polls, and request voice calls, while streamers can manage broadcasts by starting live sessions, editing details, moderating chat, and engaging followers with notifications. Moderators support streamers by managing chat interactions and polls, ensuring a respectful environment.

The platform's **AI-powered rooms** enable listeners to influence broadcasts by requesting specific songs, skipping tracks, and participating in AI-generated polls. The AI dynamically adapts content based on listener preferences, maintaining an evolving and interactive experience.

Non-functional requirements ensure scalability, reliability, and performance. The system supports thousands of concurrent users with minimal latency and achieves high uptime through AWS infrastructure. User data is protected with **GDPR** [7] and **KVKK** compliance, while TLS 1.3 [6] encryption secures all transactions. Accessibility is enhanced through WCAG 2.1 [9] standards, making the platform inclusive for all users.

By integrating third-party tools like Spotify API for licensed music, GPT-4 [8] for AI-driven content, and AWS for cloud hosting, Fennec Radio delivers a robust, secure, and scalable platform. Its blend of user customization, AI features, and reliable performance ensures a seamless experience for listeners and broadcasters alike.

3.2 Functional Requirements

- User Authorization and Authentication
  - The user can register/login using their email.
  - The user can register/login using Cognito.
  - The user should verify their email to activate their account.
  - The user can reset their password using their registered email.

- User/ Channel Customization - Profile Page
  - The user is allowed to choose their username in account creation.
  - The user can change their username from the profile once in two months
  - The user can update their profile/channel pictures from the profile page.

- Main Page
  - The user can browse and listen to live radio broadcasts from various channels.
  - The user can search for broadcasts by topic, genre, keyword, or broadcaster name.
  - The user can browse through the following channels.
  - The user can hide a certain channel to not receive the channel information.

❖ Below requirements for Broadcast Room

- Listener
  - The listener can access the streams by clicking on the channel information.
  - The listener can adjust the volume of the stream from the broadcast room.
  - The listener can adjust the quality of the stream from the broadcast room.
  - The listener can ask the streamer to join the stream via voice call.
  - The listener can participate in the polls created by the broadcaster/moderators.
  - The listener can report a radio channel or another user for inappropriate behavior.
  - The listener can follow the channel to get a notification when the streamer goes live.
  - The listener can access help resources or contact customer support.

- Streamer
  - The streamer can adjust the input sound of the stream.
  - The streamer can start live broadcasts using the go-live button.
  - The streamer can manage their broadcasts, including editing titles, banners, descriptions, and tags.
  - The streamer can create polls for listeners to participate.
  - The streamer can timeout/ban listeners from the channel's live chat.
  - The streamer can promote a listener to a moderator role in their channel.

- The streamer can unban other listeners.
- The streamer can ask a listener to connect the broadcast via voice call.
- The streamer can notify his followers.
- The streamer can access help resources or contact customer support.

➢ Moderator.
- The moderator can edit the title of the session.
- The moderator can timeout/ban other listeners.
- The moderator can create polls for users to participate.
- The moderator can unban other listeners.

❖ Below requirements for AI rooms:

➢ Listener
- The listeners can interact with chat to manipulate AI broadcasts.
- The lister can ask AI to skip a song.
- The listener can ask AI to play a certain song or certain type of song.
- The listener can participate in AI-created polls to adjust the broadcast content.

➢ AI
- The AI should play according to listeners' interests.
- The AI should alter the broadcast within its interaction with the user.
- The AI should create random polls/games to interact with the user.

3.3 Non-functional Requirements

Non-functional requirements define constraints on system performance, usability, scalability, and other qualitative aspects of the Fennec Radio platform. The main areas of focus include:

- Performance
    - The platform should support at least thousands of concurrent users during peak times with a latency of under seconds for all major interactions, such as streaming or AI-based recommendations.
    - Real-time audio processing and AI personalization must complete in under milliseconds if possible.
- Scalability:
    - The architecture must handle a highness in user traffic without service degradation by employing scalable AWS infrastructure and load balancing [4].
- Reliability:
    - The system should achieve high uptime, ensuring availability even during maintenance periods with minimal disruptions.
- Usability:
    - All functionalities must be accessible within 3 user interactions.

    - The user interface should conform to WCAG 2.1 [9] standards to ensure accessibility for individuals with disabilities.
- Security:
    - User data must comply with GDPR [7] and KVKK standards.
    - Secure streaming and transactions must be implemented using TLS 1.3 [6] encryption.
- Maintainability:
    - System logs and monitoring must allow identification and resolution of critical issues within minutes.

3.4 Pseudo Requirements

Pseudo requirements are additional constraints or assumptions imposed on the system that are not directly functional or non-functional but are necessary due to external factors. These requirements often stem from specific technologies, tools, platforms, or limitations that the project must adhere to. For Fennec Radio project, the pseudo requirements are as follows:

- Integration with Spotify API
    - The system relies on Spotify Web API to fetch and play music content. This imposes a requirement to adhere to Spotify's licensing agreements, terms of service, and API rate limits.
- Premium Spotify Account Requirement
    - Users must have an active Spotify Premium subscription to access and stream music via the Spotify Web Playback SDK due to API limitations.
- Platform and Device Compatibility
    - The system must support modern web browsers and mobile platforms (iOS and Android), ensuring compatibility with the Spotify SDK, and the application’s backend.
- Cloud Hosting on AWS
    - All backend services and data storage must be hosted on Amazon Web Services (AWS). This includes using AWS EC2 for API hosting and AWS S3 for storing audio files [4].
- Third-Party Services Dependency
    - GPT-4 for text summarization.
    - OneSignal for push notifications.
    - RevenueCat for managing subscriptions.
- Data Privacy and Compliance
    - The system must comply with GDPR [7] and other data privacy regulations to ensure user data security and privacy when storing preferences, feedback, and interactions.
- Performance Constraints

    - Due to real-time streaming requirements, the system must minimize latency to ensure seamless music playback and instant AI interaction.
- Internet Dependency
    - The platform will require an active internet connection for all its features to function, including streaming music, interacting with the AI DJ, and accessing external APIs.
- Open Source and Commercial Tools
    - The project will incorporate a mix of open-source tools (Node.js, WebRTC) and commercial services (Spotify API, AWS), which may impose licensing and cost constraints.

3.5 System Models

3.5.1 Scenarios

➢ **User Authorization and Authentication**

- **Use-case Name**: User Login and Registration
- **Actor**: User, System
- **Entry Condition**: User accesses the system for registration or login.
- **Exit Condition**: User is successfully logged in or account is created.
- **Flow of Events**:
    1. The user provides their email for login or registration.
    2. The system sends a verification email for new account activation.
    3. The user verifies the email to activate the account.
    4. The user logs into the system successfully.
- **Use-case Name**: Password Management
- **Actor**: User, System
- **Entry Condition**: User selects the reset password option.
- **Exit Condition**: User resets their password successfully.
- **Flow of Events**:
    1. The user provides their registered email.
    2. The system sends a password reset link to the email.
    3. The user resets the password using the provided link.

➢ **User Profile Management**

- **Use-case Name**: Username and Profile Management
- **Actor**: User
- **Entry Condition**: User accesses the profile page.
- **Exit Condition**: Profile is updated successfully.
- **Flow of Events**:
    1. The user updates their profile picture.
    2. The user changes their username (restricted to once in two months).
    3. The system saves the changes and updates the profile information.

- **Listening to Broadcasts**

  - **Use-case Name**: Listen to Broadcasts
  - **Actor**: Listener
  - **Entry Condition**: Listener selects a radio channel from the main page.
  - **Exit Condition**: Listener successfully listens to the stream.
  - **Flow of Events**:
    1. The listener browses available channels or searches for specific ones.
    2. The listener clicks on the desired channel to access the stream.
    3. The listener adjusts the volume or stream quality as needed.

  - **Use-case Name**: Channel Following and Notifications
  - **Actor**: Listener, System
  - **Entry Condition**: Listener accesses a radio channel.
  - **Exit Condition**: Listener receives notifications for future broadcasts.
  - **Flow of Events**:
    1. The listener selects the "Follow" option for a channel.
    2. The system registers the listener as a follower.
    3. The system sends notifications when the streamer goes live.

- **Broadcast Management (Streamer)**

  - **Use-case Name**: Start a Live Broadcast
  - **Actor**: Streamer
  - **Entry Condition**: Streamer accesses their broadcast management panel.
  - **Exit Condition**: Live broadcast is successfully started.
  - **Flow of Events**:
    1. The streamer adjusts input settings for the broadcast.
    2. The streamer clicks the "Go Live" button to start streaming.
    3. The system starts broadcasting the stream to the listeners.

  - **Use-case Name**: Broadcast Content Management
  - **Actor**: Streamer
  - **Entry Condition**: Streamer accesses the live broadcast settings.
  - **Exit Condition**: Broadcast content is updated successfully.
  - **Flow of Events**:
    1. The streamer edits the broadcast title, description, and tags.
    2. The streamer creates a poll for listener participation.
    3. The system displays the updated content to listeners.

- **Broadcast Room Moderation**

  - **Use-case Name**: Manage Listeners in Live Chat
  - **Actor**: Moderator
  - **Entry Condition**: Moderator accesses the live chat in a broadcast room.

- **Exit Condition**: Listeners are managed successfully.
- **Flow of Events**:
    1. The moderator identifies inappropriate behavior in the chat.
    2. The moderator times out or bans the listener.
    3. The moderator unbans the listener if necessary.

➢ **<u>AI Broadcast Interaction</u>**

- **Use-case Name**: Request AI Interaction
- **Actor**: Listener, AI
- **Entry Condition**: Listener accesses the AI broadcast room.
- **Exit Condition**: Listener's request is successfully handled by the AI.
- **Flow of Events**:
    1. The listener asks the AI to skip or play a certain song.
    2. The AI processes the request and adjusts the broadcast content.
    3. The listener participates in AI-generated polls to influence the broadcast.

3.5.2 Use-Case Model

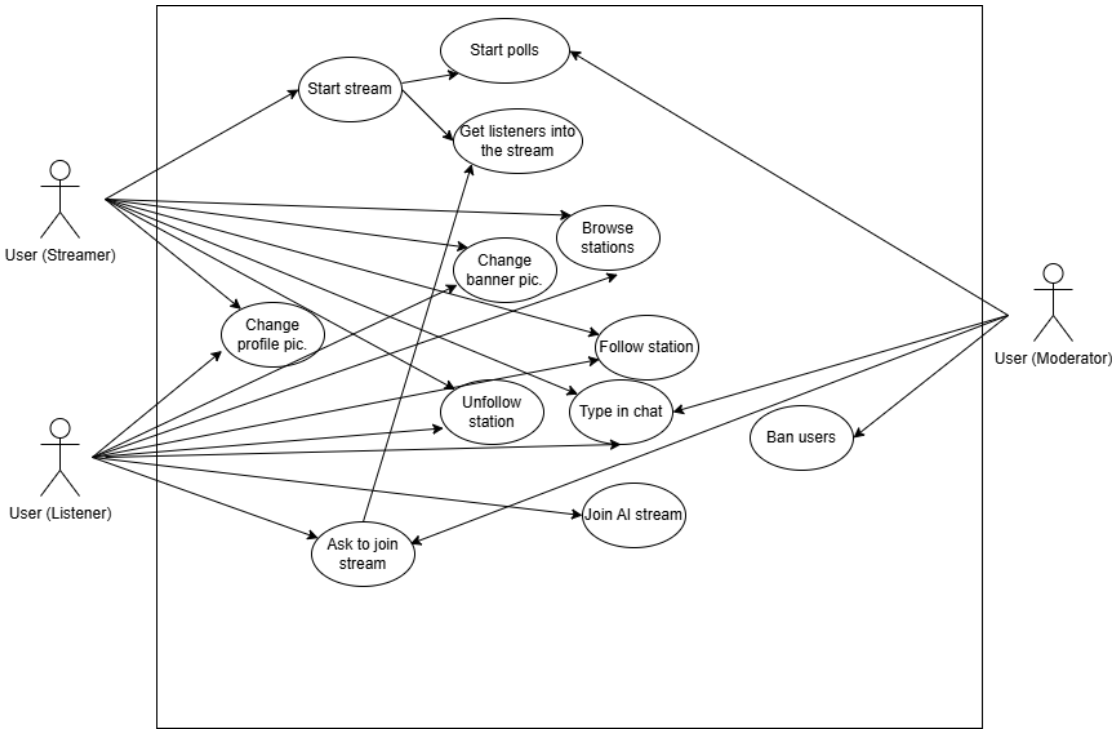

Figure 1: Use-Case Model of FennecRadio

This diagram represents a use-case model for the Fennec Radio platform, illustrating the interactions between key user roles—Listeners and Streamers—and the system. Streamers can perform actions like starting a stream, starting polls, and managing listeners by attracting them to the stream or moderating activities (e.g., banning

users). They also have options to customize their broadcasts by changing banner pictures. On the other hand, Listeners can engage with the platform by browsing stations, following/unfollowing stations, participating in chats, and requesting to join streams. They also have the ability to interact with AI-powered streams by joining AI broadcasts. Additionally, Listeners can personalize their profiles, such as by changing their profile pictures. This model highlights the user-centric and interactive features of the platform, ensuring active engagement between broadcasters and their audience.

3.5.3 Object and Class Model

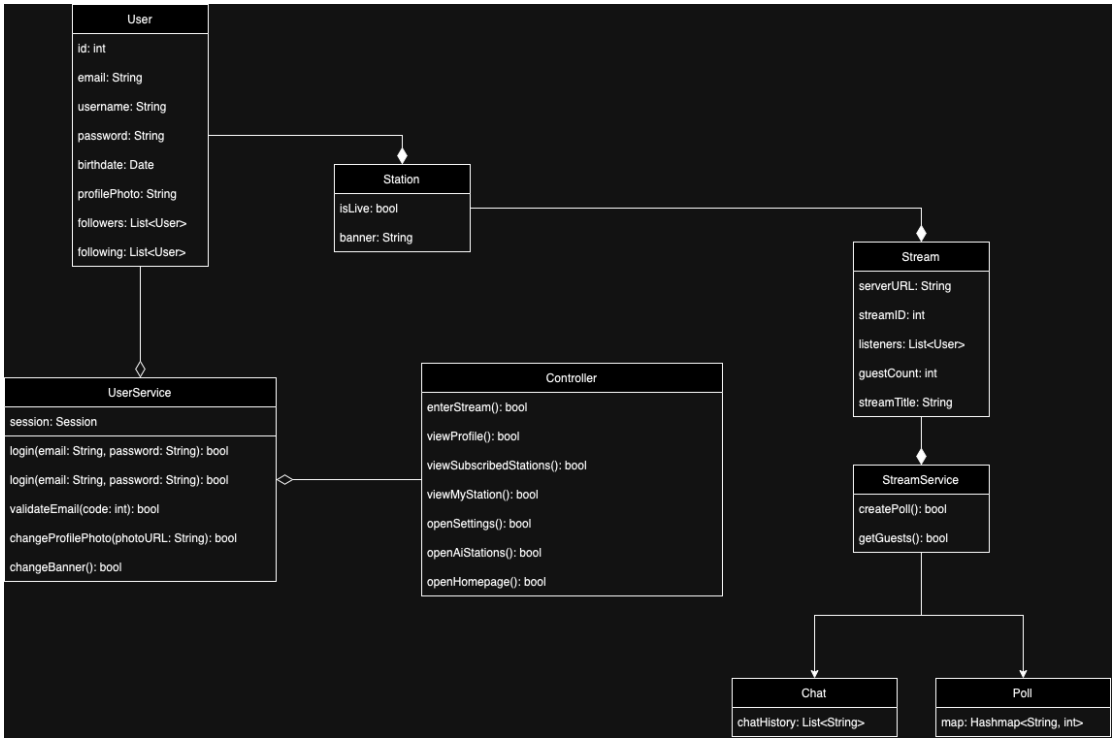

Figure 2: Class-Model Diagram of FennecRadio system

The Fennec Radio class diagram shows how users, stations, and streams are connected. Users can create or join Streams under their Stations, where they can chat and vote in polls. The UserService handles logging in and profile updates, while the StreamService manages polls and guests. A Controller helps users move between streams, profiles, and settings. The system is simple and allows users to stream, chat, and interact easily.

3.5.4 Dynamic Models

3.5.4.1 Activity Diagram

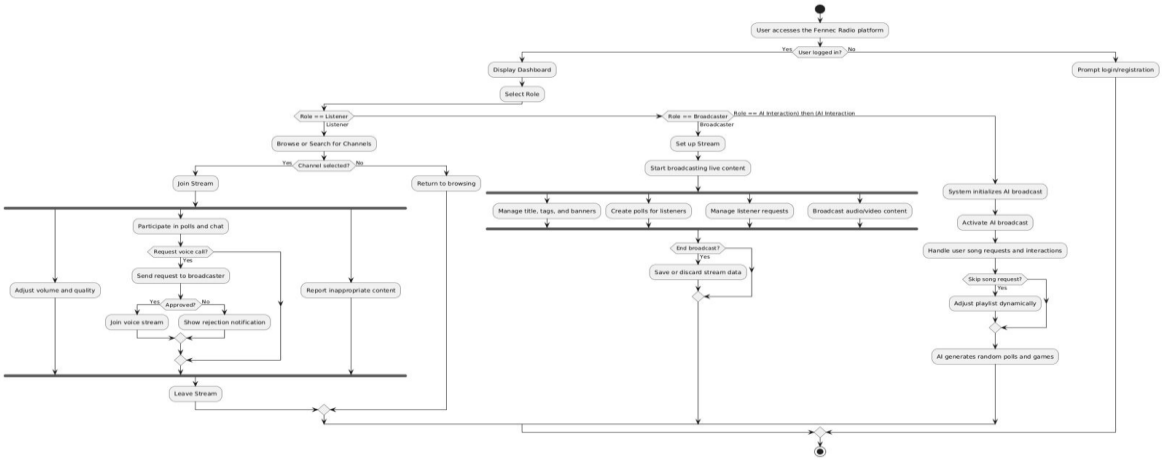

Figure 3: Activity Diagram of FennecRadio

The activity diagram shows the Fennec Radio system workflow based on user roles: Listener, Broadcaster, or AI Interaction. After logging in, users select their role. Listeners can browse channels, join streams, and interact via chat, polls, or voice requests. Broadcasters set up streams, manage content, and engage listeners. AI broadcasts, however, are automatically initialized by the system, handling song requests, playlist adjustments, and interactive polls. If users are not logged in, they are prompted to register or log in. This diagram highlights the automated nature of AI broadcasts alongside user-driven actions.

3.5.4.2 Sequence Diagram

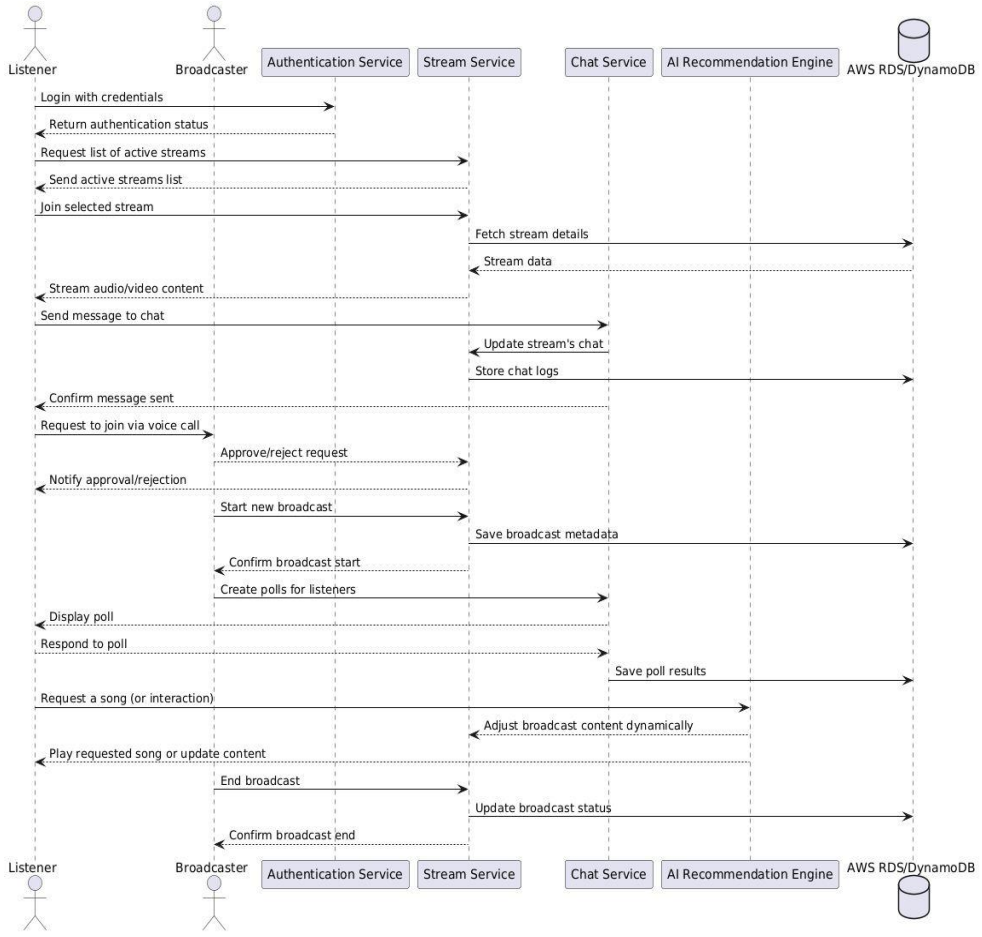

Figure 4: Activity Diagram of FennecRadio

The sequence diagram shows the interactions between Listeners, Broadcasters, and system components like Authentication Service, Stream Service, Chat Service, AI Recommendation Engine, and AWS RDS/DynamoDB. Listeners authenticate, join streams, send messages, interact with polls, and request voice calls. Broadcasters manage streams, approve requests, create polls, and end broadcasts. The AI Recommendation Engine handles song requests and adjusts content dynamically. Data like stream metadata, chat logs, and poll results are saved in AWS RDS/DynamoDB, ensuring real-time synchronization and content management.

3.5.5 User Interface

Current Fennec Radio web application UI is given below.

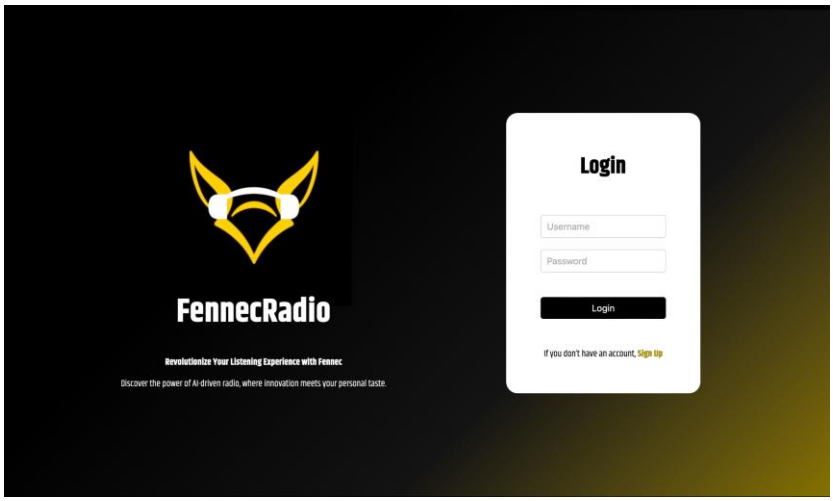

Figure 5: Login page UI

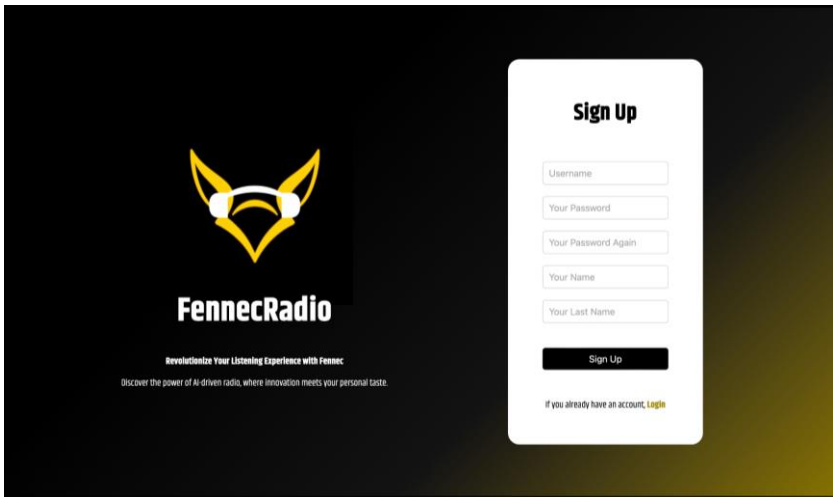

Figure 6: Sign-Up page UI

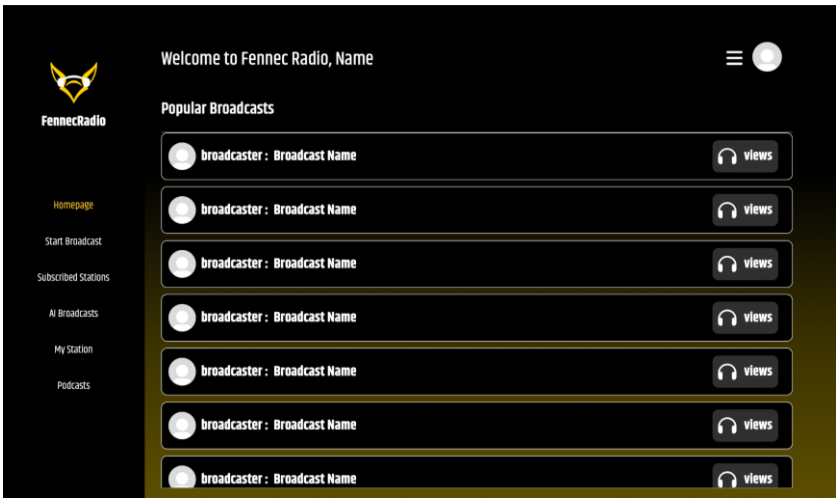

Figure 7: Homepage UI

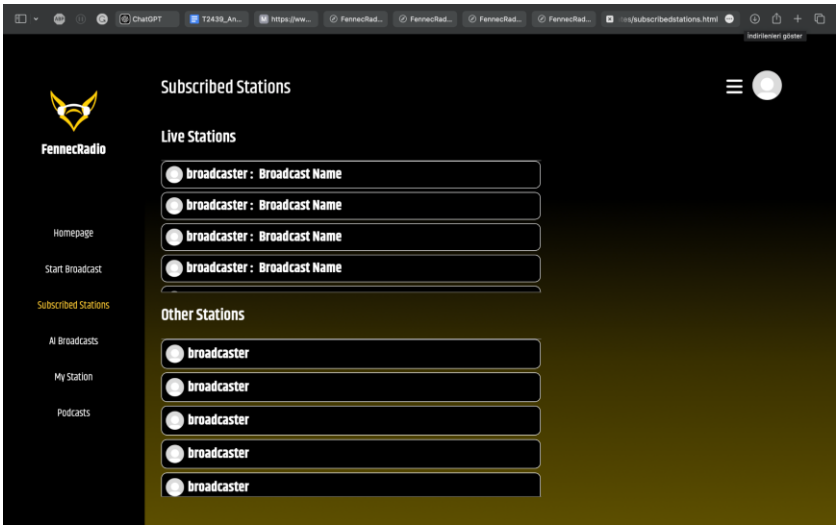

Figure 8: Subscribed Stations page UI

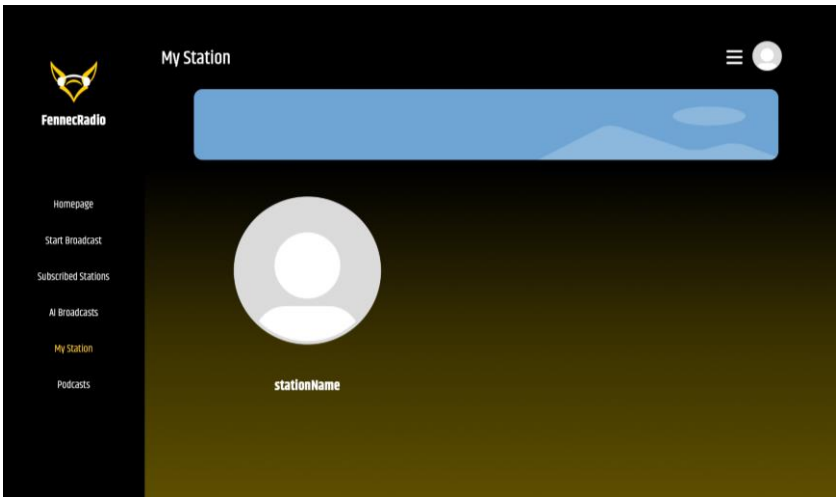

Figure 9: My Station page UI

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Constraints

● Use GitHub for version control to enable collaborative development and code management.

● Use Jira for task management, WhatsApp for instant communication.

● Implement GitHub Actions for CI/CD to automate testing and deployment processes.

● Deploy applications using containerization with Docker and orchestration with AWS EC2.

● For the front-end web application, HTML and CSS will be used.

● The technology for the backend will be Node.js framework implemented with Javascript.

● For real-time interaction, Socket.IO and WebSocket will be used.

● For security APIs reCAPTCHA or Cloudflare can be used.

● Implement user authentication using AWS Cognito with OAuth 2.0 protocols.

● Use WebRTC and Selective Forwarding Unit (SFU) for encoding and delivering multiple user live audio streams.

● Utilize AWS CloudFront as a Content Delivery Network (CDN) to ensure low latency global content delivery.

● Store user information and structured data in Amazon RDS PostgreSQL.

● Store images and recorded audio files in Amazon S3 buckets.

● Store chat messages and unstructured data in Amazon DynamoDB.

● By leveraging Hugging Face Transformers, these AI broadcasters can dynamically change the stream. Pre-trained language models such as GPT and BERT will be used.

● To facilitate real-time interactions with AI broadcasters, the platform uses OpenAI Whisper for speech-to-text conversion.

● Amazon Polly is used to synthesize natural-sounding AI broadcaster voices.

● Optimize AWS resource usage to stay within the allocated budget constraints.

● Training and deploying machine learning models will be the biggest cost of this project; however, an exact cost could not be calculated due to not knowing how much training is required and potential cost reductions with brand deals and sponsorships.

● Obtaining a domain will have a cost.

● Other services and technologies like WebRTC, Github, Node.js, Figma, Docker etc. are free and some are open source.

● API’s are free until a given amount of free calling, however it is uncertain for now because of the unpredictable requirements of the project.

● The data obtained by processing the broadcasts and users’ information are confidential and won’t be shared by third parties.

● All the data will be stored in a secure place to protect user privacy.

● Temporary storage of streams for a minimum period necessary to review and analyze reported content.

● A long lasting storage for users chat logs will be stored to use in the case of a report or harmful action.

● Establish clear guidelines prohibiting unauthorized sharing of copyrighted content and implement mechanisms to detect and prevent such activities.

● Regularly audit AI models for biases and ensure diversity in training data to promote fairness and equality.

● The Ethics of the National Society of Professional Engineers will be followed by the team in the project [4].

4.1.2 Standards

In the development of the Fennec Radio platform, various internationally recognized engineering standards are followed to ensure consistency, quality, security, and compliance throughout the project lifecycle. These standards are aligned with software engineering processes, system design, documentation, and user experience. The following standards are utilized:

- Requirements Documentation: IEEE 830

The functional and non-functional requirements of the system are documented in accordance with the IEEE 830 Standard for Software Requirements Specification (SRS). [1] This standard ensures that requirements are clearly defined, complete, consistent, and traceable throughout the project lifecycle.

- System Modeling: UML 2.5.1

The system's architecture and design models, including use-case diagrams, class diagrams, and dynamic models, are developed using Unified Modeling Language (UML) 2.5.1. [2] UML provides a standardized way to visualize the system's static structure and dynamic behaviors, facilitating effective communication among team members and stakeholders.

- Software Life Cycle Processes: ISO/IEC/IEEE 12207

The software development process follows ISO/IEC/IEEE 12207, the international standard for Software Life Cycle Processes. This standard defines the processes required for software development, operation, maintenance, and decommissioning, ensuring that the project follows a structured and consistent workflow [3].

- Cloud Infrastructure: AWS Well-Architected Framework

The cloud infrastructure is designed following the AWS Well-Architected Framework, which provides best practices for high availability, security, performance efficiency, and operational excellence. This framework ensures the application is reliable and scalable when hosted on AWS EC2 and AWS S3 [4].

- Security Standards: OWASP and TLS 1.3

The system implements security practices based on OWASP Top 10 guidelines to mitigate common vulnerabilities such as SQL injection, cross-site scripting (XSS), and data breaches. [5] All communications between system components adhere to the TLS 1.3 (Transport Layer Security) protocol to ensure data confidentiality and integrity [6].

- Data Privacy and Protection: GDPR Compliance

The system adheres to the General Data Protection Regulation (GDPR) to ensure user data privacy and security. All user preferences, feedback, and interactions are processed with transparency, user consent, and data anonymization where necessary [7].

- User Interface Accessibility: WCAG 2.1

The platform complies with Web Content Accessibility Guidelines (WCAG) 2.1 to ensure accessibility for all users, including those with disabilities. This includes features like screen reader support, keyboard navigation, and sufficient contrast ratios [9].

- Coding Standards: ECMAScript (ES6+) and Airbnb JavaScript Style Guide

The backend (Node.js) development adhere to ECMAScript (ES6+) standards for JavaScript [10].

- Monitoring and Logging: ISO/IEC 27001

System logs and performance monitoring through following ISO/IEC 27001 standards, ensuring secure storage and reporting of system metrics and incident alerts [11].

- Audio Streaming and Formats: ISO/IEC 23009 DASH

Audio content streamed through Spotify API follows the ISO/IEC 23009 Dynamic Adaptive Streaming over HTTP (DASH) standard for high-quality, adaptive streaming based on network conditions [12].

4.2 Risks and Alternatives

Fennec Radio heavily relies on user interaction and usage. If there's nobody to use the app to start live streams or listen to others, it might need to be shut down as it would bring no money and would be just costly and exceed the budget. Therefore, there must be a lot of people using it. From the technical perspective, this means scaling the platform to support high traffic and ensuring real-time interactions without latency issues may challenge the underlying infrastructure, particularly during peak usage is going to be a challenge. Additionally, relying on cloud services like AWS increases operational costs, which could surpass budgetary limits if not managed effectively. Data security and privacy represent another critical risk, as any breach could damage user trust and violate regulations such as GDPR and KVKK. Ethical concerns, such as algorithmic bias in AI-powered recommendations or misuse of the platform for harmful content, could also attract scrutiny from regulators and the public. For instance, when an inappropriate stream takes place, it needs to be taken down and given information to authorities as soon as possible such as a log of the chats or a save of that stream. Market risks include competition from well-established platforms such as Twitch, YouTube Live, and Kick, which may limit user adoption and growth. Finally, unforeseen challenges, like changes in legal requirements or dependency on third-party APIs, could disrupt development or service continuity.

To mitigate these risks, several alternative strategies can be considered. For scaling challenges, the platform can adopt a hybrid cloud approach, combining on-premise resources with cloud services to optimize cost and performance. For cost management, the team could explore free or low-cost alternatives to AWS services, such as DigitalOcean or self-hosted open-source solutions like Nginx and MinIO. Even now, the team uses free live real time communication tool WebRTC instead of Amazon IVS or Elemental Media Package to get more value from less sources. Regarding data security, implementing advanced encryption protocols and regular penetration testing can reduce the likelihood of breaches. Algorithmic fairness can be ensured by diversifying the training datasets for AI models and conducting routine audits to minimize bias. In order to become a safe and trusted platform, the project will follow RTÜK's "Platform Klavuzu" report and change the designs if needed. To counteract competitive threats, Fennec Radio can focus on niche markets or develop unique features, such as AI-driven audio personalization, to differentiate itself. Besides, Fennec Radio can follow going big in local sectors and markets then make partnerships with bigger firms.

4.3 Project Plan

	Effect level	Effect
Public health	6	Users should be careful about not becoming addicted and high-volume
Public safety	3	There might be dangerous streams but Fennec will try their best to prevent
Public welfare	7	More content, more radio listening leads to happier people and maybe a new job for some
Global factors	4	Streams can link people together around the world
Cultural factors	3	Fennec might create a new radio listening culture
Social factors	5	There can be new jobs using Fennec and people might socialize better with talking about famous streams
Environmental factors	1	No effect, web application
Economic factors	8	If things get bigger, Fennec can bring value for a lot of people such as developers, streamers, admins etc. And might get investments from abroad media or tech companies.

Table 01: Factors that can affect analysis and design

	Likelihood	Effect on the project	B Plan Summary
Risk 1: RTÜK regulations	Likely	According to RTÜK's guide report about live-streaming services, the design of the security and bandwidth values might change.	No plan B, have to commit what RTÜK asks in order to go into the real-world market.
Risk 2: AI	Medium-likely	AI streams might not be able to remix the music as the user wanted.	Make users choose their music to their liking again but not let them remix until a professional team of AI engineers come to board.
Risk 3: Stream quality	Medium-likely	Stream quality must be high, efficient and fast for all users.	The systems for streaming in the backend and database part can be improved.
Risk 4: Inappropriate streams	Not likely	Some users may try to stream dangerous/unwanted content.	Save the streams and chat logs, ban the user and give all needed information to authorities to block any of these types of actions.

Table 02: Risks

WP#	Work package title	Leader	Members involved
WP1	CS491 Reports + Website	-	Yağız, Kaan, Bartu, Feza, Atakan
WP2	CS491 Demo	-	Yağız, Kaan, Bartu, Feza, Atakan
WP3	ML and AI model	Atakan	Atakan, Feza, Bartu, Yağız
WP4	UI Design	Kaan	Kaan
WP5	Frontend Implementation	Kaan	Kaan, Yağız
WP6	Database Architecture	Bartu	Bartu,
WP7	Backend Implementation	Yağız	Yağız, Kaan, Bartu, Feza, Atakan
WP8	CS492 Reports and Final Project	-	Yağız, Kaan, Bartu, Feza, Atakan

Table 03: List of work packages

WP 1: CS491 Reports + Website			
Start date: *14.10.2024* **End date:** *16.12.2024*			
Leader:	-	**Members involved:**	Yağız, Kaan, Bartu, Feza, Atakan
Objectives: *Objective is to deliver all required reports and create the website for showcasing all of these reports including the team members and brief project description.*			
Tasks: ***Task 1.1 Website*:** Create the website of the project. ***Task 1.2 Project Information Form*:** Name the team and project, write a well-defined project summary and find an Innovation Expert and Supervisor before submitting the form. ***Task 1.3 Assessment of Innovation Form:*** Add the team ID, the team name and the team member names together with the name of the Supervisor. Have the Innovation Expert fill in the form, give a Final Expert Score and sign the form with his name and date, submit the form. ***Task 1.4 Project Specification Document:*** The project specifications report should provide a brief description and requirements of the proposed project. ***Task 1.5 Analysis and Requirements Report:*** The analysis and requirement report contains a detailed analysis of the problem. It should address all relevant issues. An analysis document of a project is produced as a result of the analysis of the system to be developed.			
Deliverables ***D1.1:*** *Website* ***D1.2:*** *Project Information Form* ***D1.3:*** *Assessment of Innovation Form* ***D1.4:*** *Project Specification Document* ***D1.5:*** *Analysis and Requirements Report*			

WP 2: CS491 Demo			
Start date: *16.09.2024* **End date:** *23.12.2024*			
Leader:	*Yağız*	**Members involved:**	Yağız, Kaan, Bartu, Feza, Atakan
Objectives: *Deliver the demo with small or zero flaws.*			
Tasks: ***Task 2.1 Show frontend*:** *Mostly from the application and unfinished pages from the Figma.* ***Task 2.2 Show live stream and backend interactions* :** *Show the live streaming mechanic in a different module at worst case and show auth, login, register etc.*			
Deliverables ***D2.1:*** *Pages of the application* ***D2.2:*** *Live streaming side project or combined with the main project.*			

WP 3: *ML and AI model*			
Start date: *04.12.2024* **End date:** *2025 Mid-May*			
Leader:	*Atakan*	**Members involved:**	*Atakan, Feza, Bartu, Yağız*
Objectives: *Create the working, interactive AI radio DJs.*			
Tasks:			

Task 3.1 ML side**:** *With machine learning, teach the AI model what to do and test.* ***Task 3.2 Implementation side*****:** *Implement the working model into the whole project.*
Deliverables ***D3.1:*** *AI model* ***D3.2:*** *Become accessible in the project*

WP 4: *UI Design*			
Start date: *15.11.2024* **End date:** *23.12.2024*			
Leader:	*Kaan*	**Members involved:**	*Kaan*
Objectives: *Design the projects frontend in Figma*			
Tasks: ***Task 4.1 Figma sketches*****:** Designing the applications frontend with chosen color palette and fonts.			
Deliverables ***D4.1:*** *A Figma collection*			

WP 5: *Frontend Implementation*			
Start date: *19.11.2024* **End date:** *2025 Mid-May*			
Leader:	*Kaan*	**Members involved:**	*Kaan, Yağız*
Objectives: *Objective is to finish the projects frontend side, all connected with database and backend*			
Tasks: ***Task 5.1 Creating pages*****:** *Create all pages (HTML/CSS) according to designs and add frameworks if needed.* ***Task 5.2 Connecting with backend*****:** *Connect all frontend codes with backend and database, change structure if needed.*			
Deliverables ***D5.1:*** *Responsive, modern and professional frontside* ***D5.2:*** *Functional buttons and translations in pages*			

WP 6: *Database architecture*			
Start date: *01.12.2024* **End date:** *2025 Mid-May*			
Leader:	*Bartu*	**Members involved:**	*Bartu,*
Objectives: *Create and match all database components with the backend. Additionally, connect AWS to the application.*			
Tasks: ***Task 6.1 Creating database components*****:** *Create all needed database elements after discussing with the team.* ***Task 6.2 Connecting with AWS*****:** *Connect to AWS and apply for more if needed.*			
Deliverables ***D6.1:*** *Database (maybe new additions are needed later on as the project evolves)* ***D6.2:*** *AWS screens that are clean and connected*			

WP 7: *Backend Implementation*			
Start date: *04.12.2024* **End date:** *2025 Mid-May*			
Leader:	Yağız	**Members involved:**	Yağız, Kaan, Bartu, Feza, Atakan
Objectives: *Create the whole backend of the project.*			
Tasks: ***Task 7.1 Live streaming protocol*: *Using WebRTC and other libraries to successfully create live streams with actual users. Implement chat and combine it with the main project.* ***Task 7.2 Connecting components with frontend* : *Connecting all backend with frontend to achieve functional website* ***Task 7.3 Bug fixes and tests*:** Tests on Postman and search for potential exploits, bug fixes.			
Deliverables ***D7.1:*** *Live stream mechanic* ***D7.2:*** *Functional pages, buttons etc.* ***D7.3:*** *Logs and reports*			

WP 8: CS492 Reports and Final Project			
Start date: *2025 End-Jan* **End date:** *2025 Mid-may*			
Leader:	-	**Members involved:**	Yağız, Kaan, Bartu, Feza, Atakan
Objectives: *Completing total of the project and deliverables*			
Tasks: ***Task 8.1 More reports*:** *Complete the remaining of the reports for the spring term and attend all meetings.* ***Task 8.2 Full project*:** *Finish the project until May or mid-May.*			
Deliverables ***D8.1:*** *Reports and meetings acknowledged* ***D8.2:*** *Fennec Radio*			

WP #	Sep. 2024		Oct. 2024		Nov. 2024		Dec. 2024		Jan. 2025		Feb. 2025		Mar. 2025		Apr. 2025		May. 2025	
WP1																		
WP2																		
WP3																		
WP4																		
WP5																		
WP6																		
WP7																		
WP8																		

Table 04: Gantt chart of WP's

4.4 Ensuring Proper Teamwork

Effective teamwork is critical to the success of the Fennec Radio project. To ensure collaboration, efficiency, and accountability, the team employs a combination of structured planning, role allocation, and agile project management methodologies.

We utilize the Scrum framework as part of agile project management to ensure iterative development, transparency, and flexibility. This methodology helps the team adapt to changes and deliver incremental value. Development is organized into two-week sprints in Jira. Each sprint begins with a Sprint Planning Meeting where the team defines goals, selects tasks from the product backlog, and determines priorities. The Sprint Backlog includes specific user stories, tasks, and deliverables to be completed during the sprint.

Sprint Retrospective, at the end of each sprint, a Sprint Retrospective is conducted to reflect on the process. What went well, identify successes during the sprint. What went wrong, discuss challenges and bottlenecks. Improvements, suggest actionable steps to improve the process in future sprints.

We utilize Scrum tools such as: Jira for task and sprint management. GitHub for version control and code reviews. WhatsApp for team communication and real-time updates. Zoom for meetings.

Team Member	**Role**	**Responsibilities**
Süleyman Yağız Başaran	Full-stack, Database	Create live-streaming protocol, mail instructors, distribute report and demo responsibilities, help frontend and implement backend
Bartu Özyıldırım	Database, Backend, AI modeling	Create database components, utilize AWS, implement backend, model development with ML
Hikmet Kaan Oktay	Full-stack	Create and design frontend, implement backend, control distribution of tasks for reports and demos
Feza Emir Çelik	AI modeling, Backend	Help implementation of backend, model development with ML
Atakan Keser	AI modeling, Backend	Help implementation of backend, model development with ML

Table 05: Member responsibility table

4.5 Ethics and Professional Responsibilities

The Fennec Radio project upholds the highest *ethical* and *professional* standards while aligning with regulatory frameworks such as RTÜK, BTK, and KVKK. The platform is committed to ensuring privacy, safety, and inclusivity, while respecting content integrity and transparency for all stakeholders [14] [15].

- User Privacy and Data Protection

All users will be informed about the data collected, and informed consent will be obtained during registration. A clear and accessible privacy policy will detail how user data is collected, stored, and used within the platform. Passwords will be stored securely using hashing techniques to prevent unauthorized access and enhance security.

- Data Retention

While long-term storage of streams will be avoided to protect privacy, temporary retention of flagged content will enable the team to address reports effectively and ensure harmful content does not go unchecked. Measures comply with KVKK and GDPR [7] guidelines to balance user privacy and platform safety.

- Compliance with Regulations

The platform strictly follows RTÜK broadcasting principles regarding privacy, public interest, and content moderation. Adherence to the BTK protocols ensures platform security and compliance with internet regulations. Clear community guidelines will outline rules prohibiting hate speech, harassment, illegal activities, and content that breaches ethical broadcasting principles. Users will have access to tools for reporting violations, and flagged content will undergo prompt review for potential removal or action [14] [15].

- Temporary Stream Storage for Moderation:

To ensure accountability, temporarily storing flagged streams will allow thorough investigation while upholding user privacy. This approach reflects the team's ethical obligation to prevent harmful content (e.g., hate speech, harassment, illegal activities) from persisting on the platform.

- Bias-Free AI Implementation:

AI features, such as content moderation and recommendations, will be audited regularly to avoid algorithmic bias and promote inclusivity. Equal opportunities for all users will be prioritized, ensuring an unbiased streaming experience.

The team adheres to software industry best practices, including secure coding standards, regular audits, and robust quality assurance processes. The platform will follow copyright laws and RTÜK regulations to prevent unauthorized use of third-party content. Reported content will be reviewed against both platform guidelines and

RTÜK regulations. Persistent violations may result in user restrictions. Measures will be taken to avoid algorithmic bias and promote equal opportunities for all users, ensuring a fair and welcoming experience. Regular communication with RTÜK and other regulatory bodies ensures ongoing compliance with legal and ethical frameworks [15].

4.6 Planning for New Knowledge and Learning Strategies

To ensure the success of Fennec Radio, the team must stay up-to-date with emerging technologies and industry standards. The learning strategy includes:

- Research and Development:
    - Allocate weekly sessions for team members to research advancements in AI, audio streaming technologies, and cloud services.
    - Stay updated on Spotify API updates and third-party service changes to maintain compliance and exploit new features.
- Team Knowledge Sharing:
    - Organize bi-weekly knowledge-sharing sessions where team members present findings from research or experiences gained during development.
    - Maintain an internal knowledge base documenting key discoveries and best practices.
- Continuous Feedback:
    - Collect feedback from early users and adapt the learning process to address real-world issues and user needs.
- Documentation and Training:
    - Ensure comprehensive documentation of system design and functionalities to facilitate smooth onboarding for new team members or external collaborators.
    - Train team members on the latest tools used, GPT integrations, and other AI-based services.

5 Glossary

KVKK: Kullanıcı Verilerini Koruma Kanunu (Personal Data Protection Law of Türkiye)

GDPR: General Data Protection Regulation of Europe

RTÜK: Radyo Televizyon Üst Kurulu (Radio and Television Supreme Council)

BTK: Bilişim Teknolojileri Kurumu (Information and Communication Technologies Authority of Türkiye)

API: Application Programming Interface

AWS: Amazon Web Services

6 References

[1] IEEE, *IEEE 830: Recommended Practice for Software Requirements Specifications*, IEEE Standards Association, Available: https://standards.ieee.org/

[2] OMG, *Unified Modeling Language (UML) Specification, Version 2.5.1*, Object Management Group, Available: https://www.omg.org/spec/UML/2.5.1/

[3] ISO/IEC/IEEE 12207, *Systems and Software Engineering – Software Life Cycle Processes*, International Organization for Standardization, 2017, Available: https://www.iso.org/standard/63712.html

[4] Amazon Web Services, *AWS Well-Architected Framework*, AWS Documentation, Available: https://aws.amazon.com/architecture/well-architected/

[5] OWASP Foundation, *OWASP Top 10 – Web Application Security Risks*, 2021, Available: https://owasp.org/www-project-top-ten/

[6] IETF, *RFC 8446 - Transport Layer Security (TLS) Protocol Version 1.3*, Internet Engineering Task Force, 2018, Available: https://tools.ietf.org/html/rfc8446

[7] GDPR, *General Data Protection Regulation*, European Union, 2018, Available: https://gdpr-info.eu/

[8] OpenAPI Initiative, *OpenAPI Specification 3.0*, OpenAPI Initiative, Available: https://www.openapis.org/

[9] W3C, *Web Content Accessibility Guidelines (WCAG) 2.1*, World Wide Web Consortium, 2018, Available: https://www.w3.org/TR/WCAG21/

[10] ECMA International, *ECMAScript 2015 Language Specification (ES6)*, ECMA, Available: https://www.ecma-international.org/

[11] ISO/IEC 27001, *Information Security Management*, International Organization for Standardization, Available: https://www.iso.org/isoiec-27001-information-security.html

[12] ISO/IEC 23009, *Dynamic Adaptive Streaming over HTTP (DASH)*, International Organization for Standardization, Available: https://www.iso.org/standard/57623.html

[13] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

[14] “Platform Klavuzu”. RTÜK. Accessed 16.12.2024

[15] “Yayın İlkeleri”. RTÜK. 2014. Accessed 16.12.2024

[16] NSPE, “NSPE Code of Ethics for Engineers,” National Society of Professional Engineers, Jul. 2019. Accessed Nov. 22, 2024. https://www.nspe.org/resources/ethics/code-ethics.